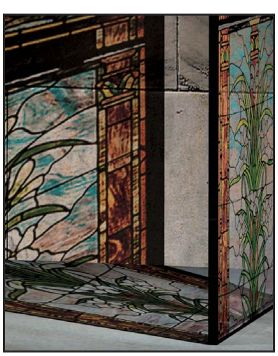
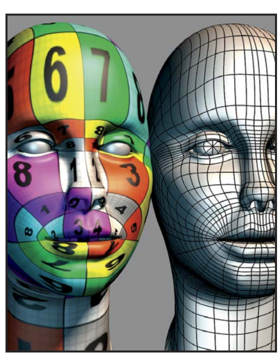
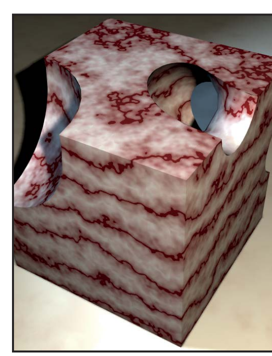
[CHAPTER TEN]

Designing and Assigning Textures

Texture mapping is the art of adding variation and detail to 3D surfaces that goes beyond the level of detail modeled into the geometry. This chapter will discuss the types of textures you can create, how to align textures with your models, and different ways to create textures.



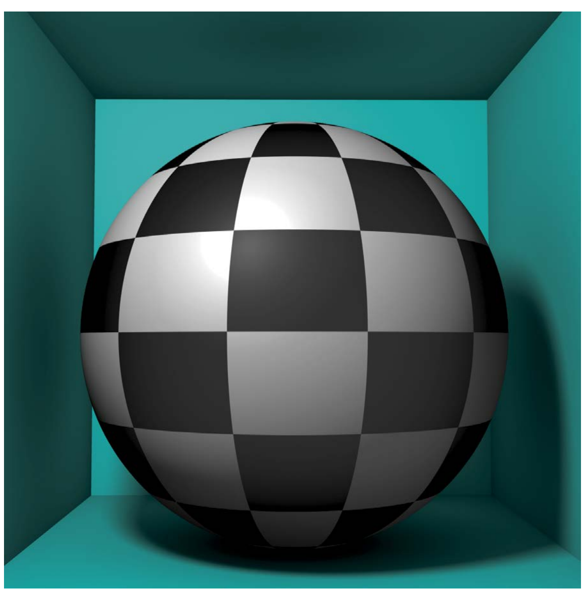
Mapping Surface Attributes

You can use textures to control many attributes of a surface, creating many different effects in your rendering. The seven most common mapping techniques are

- Color
- Specular
- Incandescence
- Transparency
- Displacement
- Bump
- Normal

Color Mapping

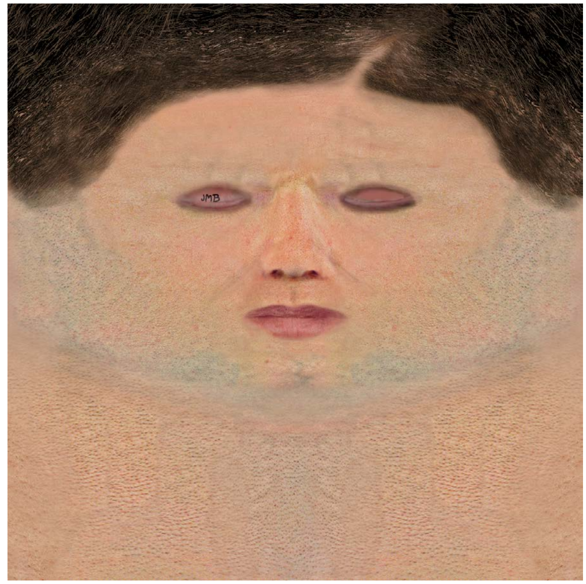
Color mapping (sometimes called *diffuse mapping*) replaces the main surface color of your model with a texture. For instance, a black-and-white grid is applied as a color map in Figure 10.1.



[Figure 10.1]
A sphere is texture mapped with a black-and-white checkerboard as a color map.

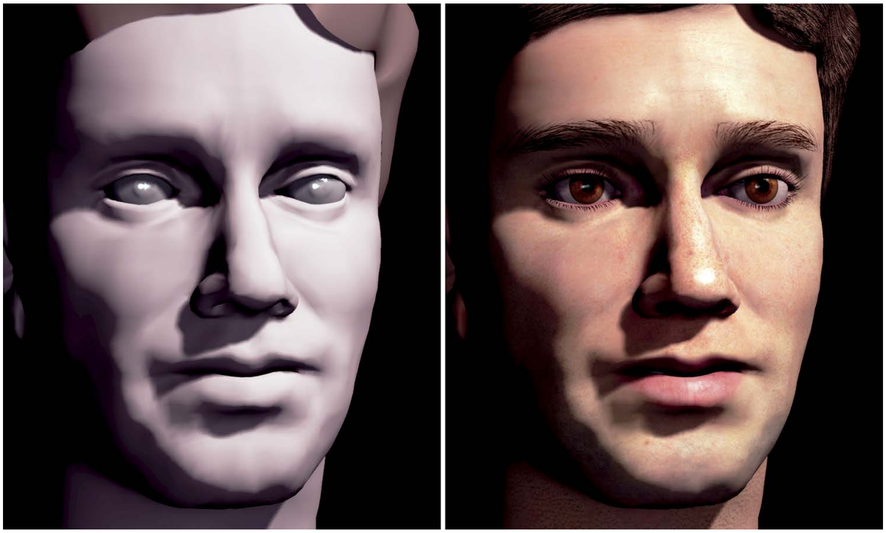
Your color map modifies the tint and intensity of the diffuse light a surface reflects. In some renderers, color mapping and diffuse mapping are listed as two different kinds of map. In this case, the tones of the color map are multiplied by the tones of the diffuse map. For example, a 50% gray in a diffuse map cuts the brightness of the color map in half.

Don't paint highlights, shadows, or lighting variation into the color maps themselves—when you render, lighting should be added by the lights. Don't texture map a dark area onto a model and expect it to look like a shadow; in the context of an animation it's more likely to look like dirt or dark paint on your model. Don't create white spots in your textures and expect them to look like highlights on the surface, because they are more likely to look like white spots painted onto the model. The best color maps usually look very flat when viewed by themselves. For example, see the color map for a man's face in Figure 10.2.



[Figure 10.2] This evenly lit texture of a face shows how color maps can avoid built-in lighting.

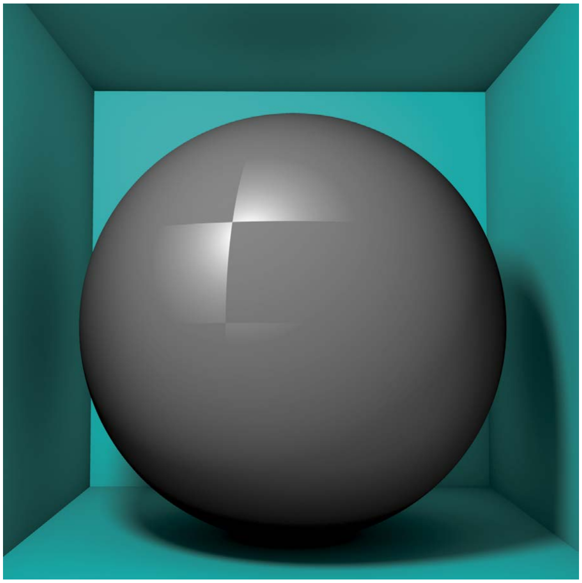
In a realistic rendering, object colors usually should not include pure black or pure white and should avoid completely saturated red, green, or blue colors. A 100% white color means that 100% of the light hitting the surface is diffusely reflected, which does not occur in the real world. Similarly, a pure black surface shows no diffuse light reflectance, which is also unrealistic. This is just a rule of thumb, but in most cases, it's a good idea to keep the red, green, and blue color values in your texture maps between 15% and 85%. Although your texture maps may look a bit flat by themselves, your final output gains contrast and directionality from your lighting, as shown in Figure 10.3.



[Figure 10.3] This head model (left) is textured with the color map from Figure 10.2 (right).

Specular Mapping

Specular mapping varies the brightness and color of the specular highlights on different parts of an object's surface. I applied a checkered pattern as a specular map around the entire object in Figure 10.4, but you can only see its influence in the area of the specular highlight.



[Figure 10.4] This sphere has a checkerboard as a specular map.

Your specular map does not create specular highlights by itself; highlights still need to come from light sources. Your specular map can tint the highlights, change their brightness, or even block them completely from a particular part of your model. However, the effects of specular mapping are only visible in places where a specular highlight would have appeared anyway.

Bright areas in your specular map make highlights brighter, creating a shinier area on your object. Dark tones of a specular map make highlights less visible, and pure black completely prevents highlights from showing up on the corresponding parts of your model. For example, Figure 10.5 shows a specular map for a man's face. White areas of the map produce a shiny forehead and nose, while dark areas prevent highlights on the stubble of his chin and cheeks.

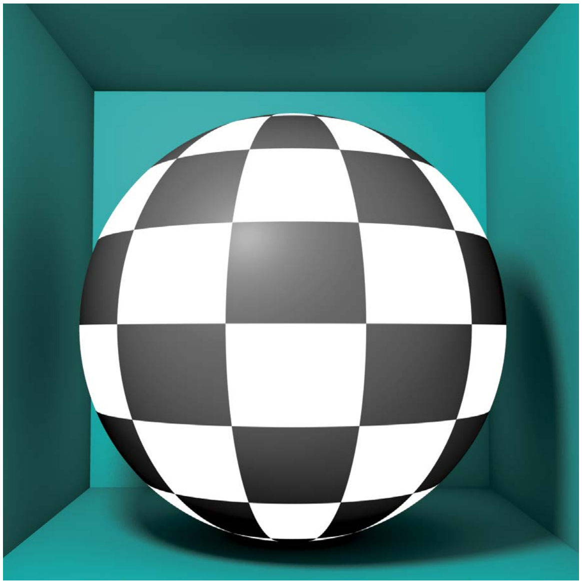
Because the effects of a specular map are visible only in areas of your model that already receive a specular highlight, you can't see the full influence of the map in one still image. To thoroughly evaluate what the map is doing all around your model, render a *turntable test*, which is an animated loop that shows the model rotating 360 degrees in front of the camera. Because the camera and lights remain in place, you see the highlights travel across the surface of the model as it spins, and you are able to spot where the highlights are brighter, where they are less bright, and which areas don't receive highlights at all.

[Figure 10.5] A specular map for a man's face shows where the face should be shiny (the bright parts) or dull (the dark parts of the map).

Reflectivity is a close relative of specularity. Because specular highlights represent the reflection of light sources, and reflectivity controls where a shader reflects everything else, you generally expect to see specularity in the same places that you see other reflections. You can use the same texture map that governs specularity to also control your reflectivity. In Maya, a map on the specular color controls both specular highlights and reflectivity by default.

Incandescence Mapping

Incandescence mapping (also called *luminosity*, *ambience*, or *constant mapping*) uses a texture map to simulate self-illuminating properties of an object. The colors in an incandescence map are added to the colors in your final rendering, without regard for your lighting. As shown in Figure 10.6, incandescence maps are visible on a surface even in shadow areas; they don't need a light source to illuminate them.



[Figure 10.6] This sphere has a checkerboard as an incandescence map.

Incandescence maps are perfect for tasks like adding lights to the side of a building or ship, adding the glowing picture to a television or monitor, or texturing the surface of a lit lightbulb. Incandescence maps are also handy if you want to combine 2D pictures with your 3D renderings, and you want the pictures to keep their own lighting, without needing lights in your scene to illuminate them.

If you apply an incandescence map to an object that has no specularity (pure black specular color) and no diffuse shading (pure black diffuse color), then it produces a flat area of color, and the exact shades and tones from the source map are output in the rendering of the object. Some renderers offer a surface shader or light source material that produces the same effect as an incandescence map. However, an incandescence map is just one type of map that you can use within a general-purpose shader, in addition to standard diffuse and specular shading. This means it's easy to combine an incandescence map with other shading effects. Parts of your incandescence map that are pure black will add nothing to the brightness of a surface, so in these areas you can see standard diffuse and specular shading that responds normally to your lighting.

In Maya, mapping the ambient color is different from mapping the incandescence, in that the ambient color gets multiplied by the main shader color before it is added to your illumination. Figure 10.7 shows an example of this. If you brighten the lamp shade with a map on the ambient color (left), the

[Figure 10.7]
An ambience map (left) is multiplied by the color map, while an incandescence map (right) is added without regard for surface color.

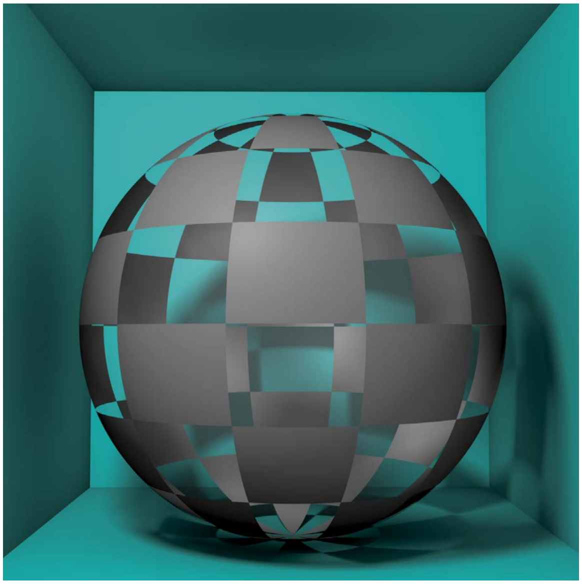
ambient color gets multiplied by the color map, and the texture is preserved and reinforced. Brightening the lamp shade with an incandescence map (right), on the other hand, makes the surface brighter but is not filtered through the color map.

Transparency Mapping

Transparency mapping has several useful functions. The simplest function of a transparency map is to simulate a partially transparent surface in which some parts are more transparent than others. You can make parts of a surface more transparent where there's a lighter color in the transparency map or less transparent where there's a darker tone.

Figure 10.8 shows a white and black checkerboard applied to a surface's transparency. Dark areas of a transparency map make a surface less transparent, and light areas make the surface clearer.

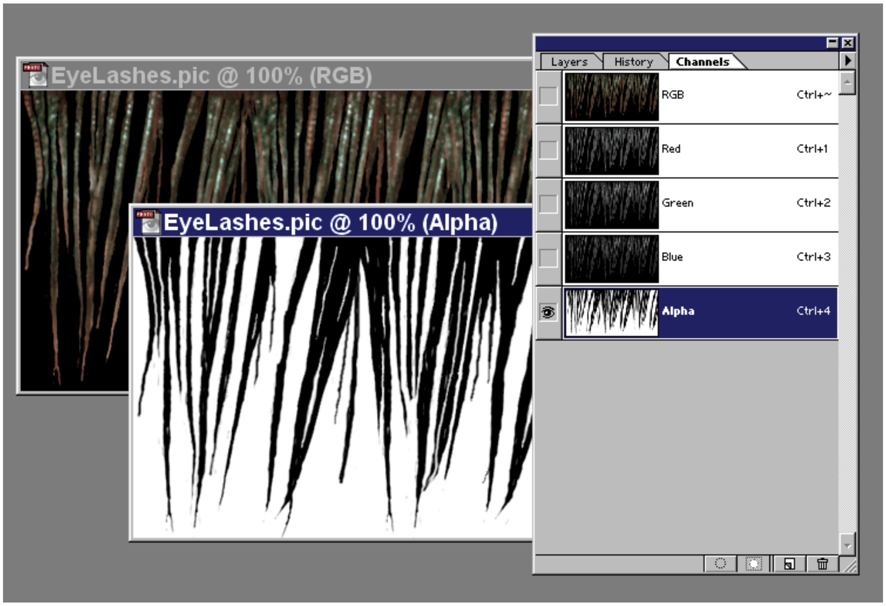
You can achieve colored transparency with a multicolored transparency map to create effects such as the stained glass window in Figure 10.9.



[Figure 10.8] This sphere is transparency mapped with a checkerboard pattern.

[Figure 10.9] A multicolored transparency map can look like a stained glass window.

You can also use transparency mapping to cut out detailed shapes and patterns from a surface. Figure 10.10 shows a transparency map designed to simulate hairs or eyelashes. Most 3D programs can use a texture map's alpha channel to determine the level of transparency in a textured surface; some people refer to transparency mapping as *alpha mapping* for this reason. The eyelash texture map contains color information, which you can apply as a color map, and also has transparency information in its alpha channel.



[Figure 10.10] An eyelash texture uses an alpha channel for transparency.

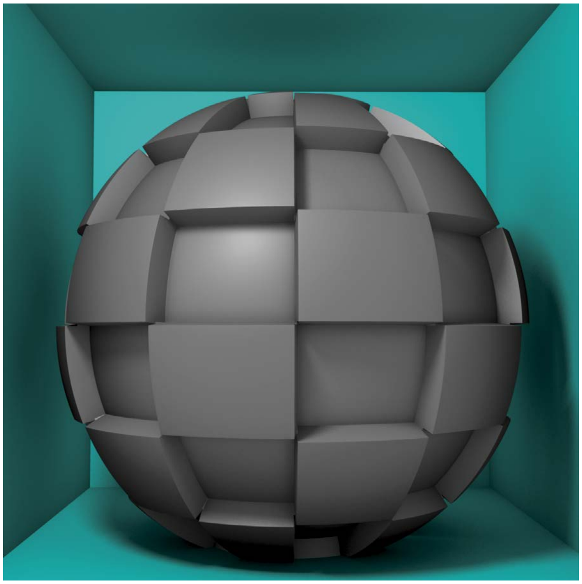
The eyelash texture creates the eyelashes and ear hairs for the hippo in Figure 10.11. The eyelash texture map creates the illusion of separate lashes by hiding parts of the eyelash surface in stripes of transparency. You can use the same technique to make a row of grass that sticks up from a lawn, the bangs of a character's hair, or a distant tree line on a hillside.

To make sure that a transparent area will really be invisible, make sure that it does not have any reflections or specular highlights on it. You can turn off reflectivity and specularity from the whole surface or map the specularity and reflectivity to make them both become 0 wherever the surface is transparent. Also make sure that you turn off refraction, or that the index of refraction is mapped to 1, to make areas completely invisible.

[Figure 10.11]
Ear hair and eyelashes are created with transparency mapping.

Displacement Mapping

A displacement map changes the shape of a surface. The brighter the tone in a displacement map, the farther out a point on the surface is displaced. Figure 10.12 shows the checker pattern displacement mapped onto a sphere. The change to the shape of the sphere is most noticeable at the edges.



[Figure 10.12]
A sphere's shape is changed by a displacement map.

Displacement height determines how far in or out a displacement map should push your surface. In many shaders, you also have an offset parameter to set how much displacement you start with where the map is pure black. You may leave the offset at 0 so that black in the displacement map has no effect, or you may make it start at a value of –0.5 so that black displaces the surface inward and a middle gray is used where you want the map to have no effect.

Your renderer makes details in your displacement map visible by moving parts of your model as they are *tessellated*. Tessellation means subdividing a surface into many small polygons at render time. You may think that the tessellation quality on a NURBS surface or subdivision surface is only adjusted to make curved models appear smoother. But once you are using a displacement map, you need to raise the tessellation quality of your models to make details in the displacement map visible. In some software, the tessellation is labeled *subdivision* or *displacement quality*. Using too little tessellation on a displacement-mapped surface can result in a blocky, jagged, or poorly defined displacement. Increasing the tessellation too high on a surface can give you a more detailed, accurate representation of the map, but it can add greatly to the memory and time you need to complete your rendering.

When using a Reyes renderer such as RenderMan, you don't need to make tessellation adjustments; as long as your *shading rate* is low enough, displacement maps are rendered accurately at any resolution. At shading rates below 1, the displacement is even smaller than a pixel. Your only other concern in rendering displacement maps in a Reyes renderer is to make sure that a shader's displacement bounds reflect the maximum height of your displacement map.

Because displacement maps don't change the shape of a surface until render time, animators do not see the displacement while posing a character. Be very careful about applying displacement to ground or character surfaces where you need accurate contact; if he doesn't see the displacement, an animator might make the character's feet cut through parts of the ground that displace upward, or float above parts of the ground that displace downward. Sometimes you may need to provide animators with a polygon mesh that represents the shape that a surface will be displaced into as reference for the appearance of the displacement map.

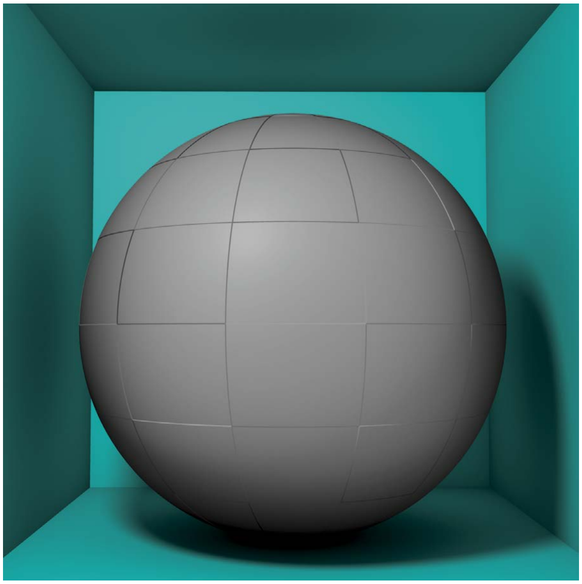
Bump Mapping

Bump mapping is a trick that simulates small details on an object's surface, without actually displacing the geometry. Figure 10.13 shows the influence of a checkerboard as a bump map. Bump mapping isn't as convincing as displacement mapping, but it can render much more quickly. You can use bump mapping to simulate a very small-scale roughness on surfaces, which would barely stick up high enough to be noticeable as a displacement map.

Bump mapping works by changing the surface's shading as if small details were present, even though no extra detail is actually added to the surface. The shading of a surface is based on an angle called the *surface normal*, which is usually perpendicular to the geometric surface of an object. A bump map changes the surface normals, making the object respond to light as if additional details were present in the geometry.

Bump maps are encoded with brighter tones representing higher elevations and darker tones representing lower elevations. A white dot represents a bump up from the surface, and a dark dot represents a depression into the surface. A transition from dark to light creates a ridge between different altitudes. People often use a neutral 50% gray as a starting point for a bump map, but any flat tone works the same way. In a bump map, an area of constant color, with no variation in shading, has no effect on a surface.

[Figure 10.13]
A bump map simulates ridges without changing the shape of the sphere.



Because bump mapping does not actually change the shape of your geometry, it has some limitations:

- The outline or silhouette of an object is not changed by a bump map and remains smooth or straight even if the map simulates a very rough surface. For example, a sphere with a bump map could still have a perfectly circular outline.
- Shadows cast by a bump-mapped object still have their original shape.

- Shadows received on a bump-mapped surface remain straight and are not distorted as if they really landed on rough ground.
- Unlike modeled or displaced details, details added to a surface via a bump map do not cast attached shadows onto the surface.
- The line where a bump-mapped object intersects with another object is not changed by a bump map and can give away the real shapes of the objects.

These limitations are all corrected with displacement mapping, because displacement actually changes a surface's shape. However, bump mapping is a useful cheat for times when you aren't bothered by these limitations. Here are the main effects that a bump map can accurately simulate:

- Diffuse shading is varied as if the bumps really existed in the surface.
- Specular highlights are broken up and scattered. Tiny highlights can even appear on individual bumps caused by a bright pixel in a bump map.
- Reflections (raytraced or reflection mapped) are distorted and broken up.
- Refraction (the view through transparent, raytraced surfaces) is correctly modified and distorted.

Of course, you can also create these effects with a real displacement map, provided that the surface was subdivided finely enough, but bump maps can simulate these effects without actually changing the shape of the surface. Figure 10.14 shows how a bump map breaks up highlights and reflections. The water surface starts out perfectly smooth and uniform with no bump map (left), but when a bump map is applied (right), the raytraced reflections are rippled and distorted. Additional highlights are also added on the right, and thanks to the bump, they are broken up into blotchy shapes instead of appearing as perfect circles. If you are going for a subtle effect such as distorting the highlights and reflections on a surface, bump mapping can be a convincing and convenient tool.

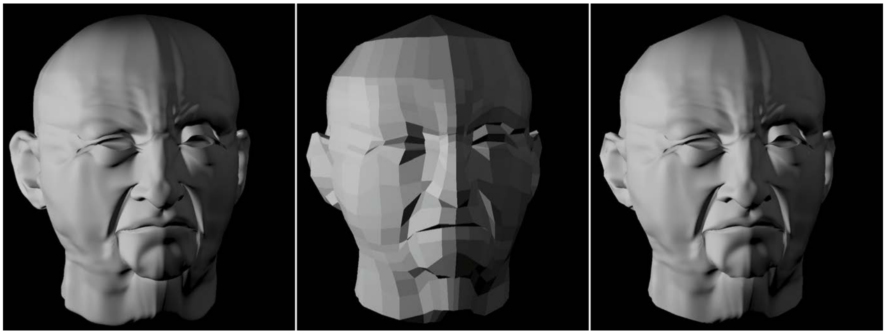
[Figure 10.14]
Water is a smooth surface (left), but adding a bump map distorts reflections and highlights (right) in this scene by Gladys Leung (www.runtoglad.com).

Normal Mapping

Normal mapping is similar to bump mapping in that it cheats your shading without actually changing the shape of the model.

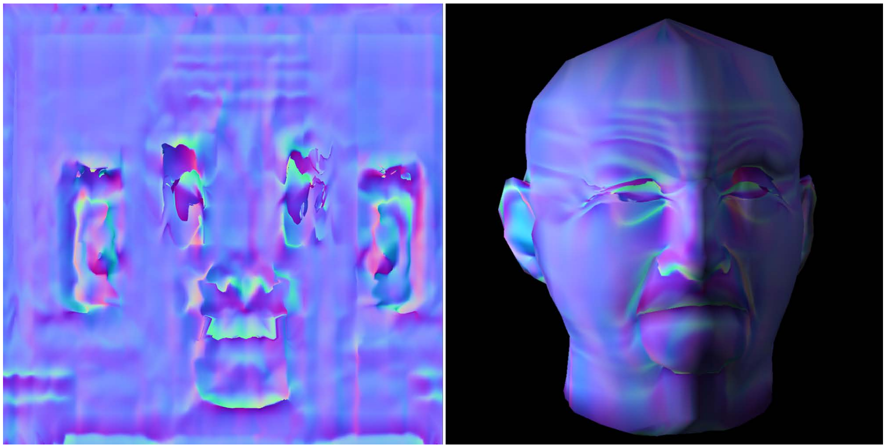
Compared to bump mapping, normal mapping is a more direct, specific way to perturb surface normals. In bump mapping, pixel brightness is interpreted as a height, and a slope derived from the heights of adjacent pixels determines the angle assigned to surface normals. In a normal map, a 3D angle for a normal is directly specified by three values per pixel, stored as three color channels in the map.

The most common use for normal mapping is to mask the difference between a high-resolution, detailed model and a low-polygon model. Figure 10.15 shows a high-resolution model (left), a simplified low-polygon model without any textures (center), and the model with a normal map (right). On the right, the model looks almost as if it were made of far more polygons, although the faceted outline of the model gives away that the geometry is still low-polygon just like the center image.



[Figure 10.15]
A high-polygon model (left) can be replaced by a low-polygon model (center) if it is made to look higher resolution with a normal map (right).

Normal maps can be computed automatically based on the difference between two versions of a model. The normal map shown in Figure 10.16 represents the difference between the low-polygon model and a higher resolution model. Normal mapping allows video game developers to use low-polygon models while they create the appearance of higher levels of detail. Normal mapping is not used as widely in film and television production, but it can be used for small background objects within complex scenes.



[Figure 10.16] This normal map represents the difference between the low-polygon and high-polygon models in Figure 10.15. On the right it is shown as a color map on the low-resolution model.

Polynomial and Other Types of Texture Mapping

Polynomial texture mapping (PTM) is another cheat to your shading, like bump mapping and normal mapping. Compared to bump mapping, PTM is a newer, more sophisticated, and more realistic way to simulate small details on a surface. Unlike bump mapping, PTM simulates the shadowing between bumps, translucency of details such as small grains of rice, and even the interreflection of light between surface features. PTM can simulate fine textures, such as the weave of a fabric or towel, including the illumination shared between fibers, in a more subtle and accurate way than displacement mapping or bump mapping.

Instead of a single grayscale image, like a bump map, a PTM starts with six channels, which you can store as a six-channel TIFF file or as two files each with red, green, and blue channels. People can acquire PTM data via light probes that digitize a real material while lighting it from many different angles with an array of strobe lights. They can also bake PTM data in 3D, based on a source displacement map with assigned material characteristics.

PTM is a published technique that is used at some studios that write their own shaders, but most users are still waiting for their software to support PTM. Hopefully PTM, or something similar, will become widely available in commercial software in the future to provide a more modern and complete alternative to bump mapping or small-scale displacement.

The number of effects that texture maps can govern is virtually unlimited. Programs may use specific maps to control almost any kind of effect. For example, some programs use a specific type of map to control the shape and color of a lens flare or the surface appearance of a particle. Even a plug-in may use its own type of texture maps for special purposes, such as determining the length, direction, and curliness of hairs emitted from a surface.

Aligning Maps with Models

How does a texture map, which is a 2D image, get wrapped around a 3D model? Which parts of the texture appear on which sides of the model? Before you begin creating a texture map, you need to think about what strategy you're going to use to align it with your geometry. There are several different approaches to this problem.

Assigning UV Coordinates

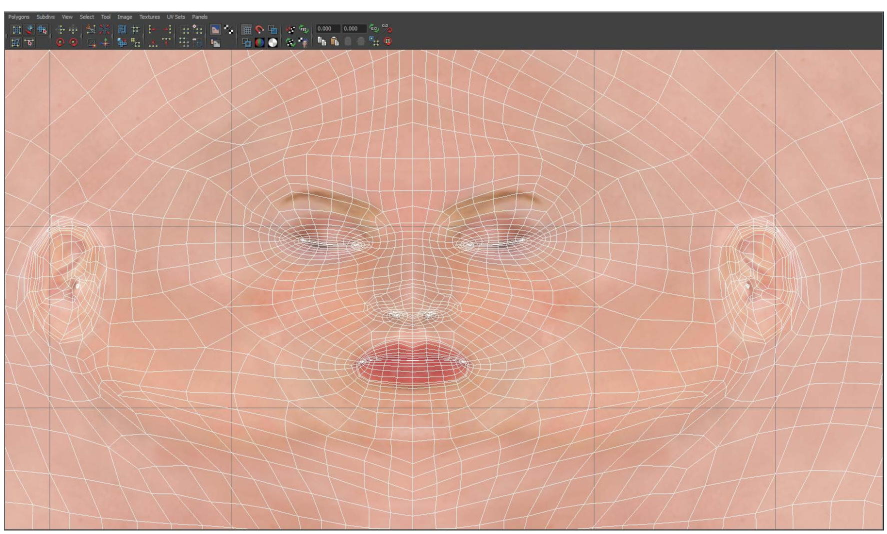
Assigning UV coordinates to a model is one of the most common approaches to aligning texture maps with geometry. By assigning unique coordinates in U and V to each point on your model, you can define exactly where each part of your map appears on the surface of the model.

Coordinates in U and V describe a position on a surface in much the same way that X and Y describe a pixel location in a texture map. (Although some people think of the V as standing for vertical, they are really named UV coordinates because U, V, and W are the three letters before X, Y, and Z.)

Wait until you are done building a 3D model and do not need to change its structure again before you assign UV coordinates to it. Adding or deleting points from the model after UVs are assigned can corrupt your UVs, or create gaps or irregularities in the mapping. If you change your model after you have assigned UV coordinates and painted texture maps, then not only could you need to create new UVs, but you may need to paint the texture maps again.

To assign new UV coordinates, you generally start with a choice of projection types that will be a basis for how the UV coordinates are distributed through space. If one of the projection types matches the overall shape of the model, you may get a great fit by projecting UV coordinates onto the model. For example, a spherical projection may be a great match for a model that is round like a piece of fruit. As another choice, you can use an automatic mapping function to make your software distribute the UV coordinates as evenly as possible over the vertices in your model so that each vertex gets a unique UV coordinate.

Most 3D programs include a UV Texture Editor window, as shown in Figure 10.17, which shows an unfolded view of the object's geometry superimposed over the map. Notice how the UV texture editor shows where on the model the eyes, ears, nose, and mouth will appear. If you move points in this view, it changes their UV coordinates, giving them a different alignment with the texture map.



[Figure 10.17] The UV Texture Editor shows an unfolded view of the polygons as they align with the texture map.

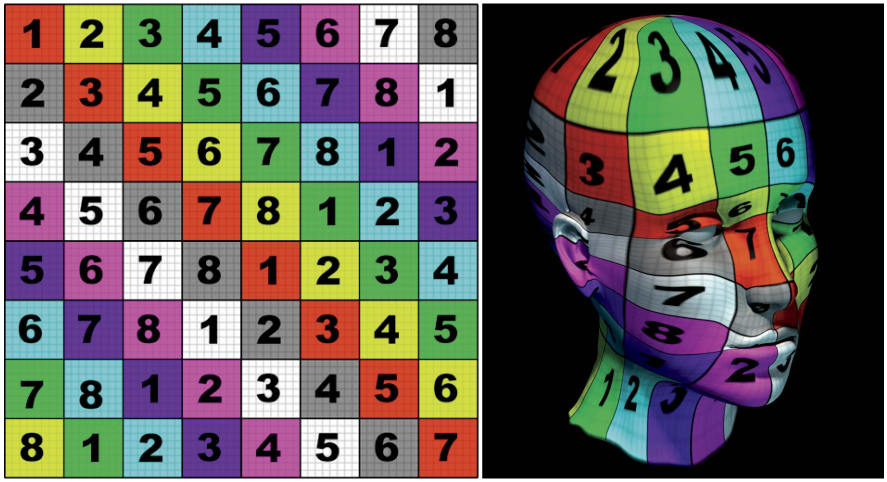
If two points on the model appear in the same place within the UV Texture Editor, that means that both points on your model have the same UV coordinates assigned to them. This can happen easily if you assign the coordinates via a projection. Projections move right through space in whatever direction you specify, and a given part of the projection can easily cross through your model two or more times, giving different parts of the model the same UV coordinates. When this happens, the same pixels from your texture map appear in two different parts of the model.

Sometimes, particularly in game development, artists intentionally have different parts of a model use the same UV coordinates. For example, a logo may appear on the front and both sides of a truck because the artist repeated the same part of the texture map onto several different faces of the model. By doing this, he is using every pixel in the texture map in the most efficient way possible. However, if you are working in motion pictures, this is not a common practice. Most of the time, you want to give each part of your model unique UV coordinates so that you'll have the creative freedom to paint unique colors or patterns onto any part of it.

Using Implicit UV Coordinates

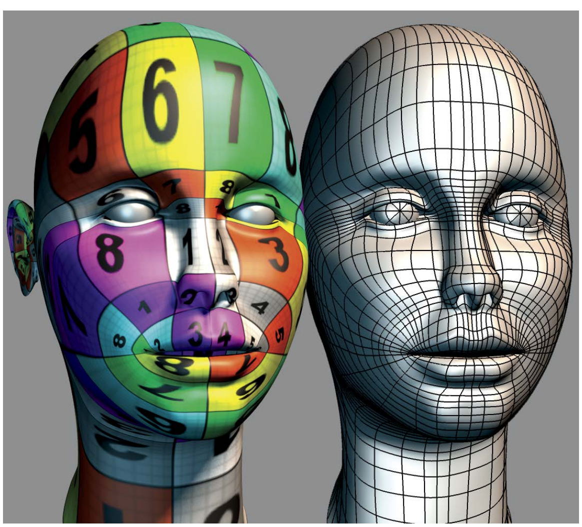
If you happen to be using NURBS models, you do not need to assign UV coordinates, because NURBS models have *implicit* (or "built-in") UV coordinates. From the moment a NURBS surface is created, it has implicit UV coordinates at each point. No matter how you edit a NURBS surface, there will always be a unique UV coordinate for each point on the surface. Figure 10.18 shows a map marked with unique colors and numbers (left) applied to the UV coordinates of a NURBS surface (right).

The built-in UV coordinates are made possible by the structure of a NURBS surface, which is always a grid. A NURBS surface is made up of a set of curves that run in a direction called U; these intersect a set of curves that run in the other direction, called V. Each point has its own UV coordinate value that describes its position within the grid structure of the surface. Even though you can bend and reshape the surface flexibly, it will always remain a grid. To preserve the grid structure, NURBS modeling software does not allow you to delete individual points; you need to delete the entire row or column of points at once.



[Figure 10.18] A map designed to show alignment (left) follows the UV coordinates of a NURBS head (right).

A map you apply to a NURBS surface follows its UV coordinates, usually with the X (horizontal) dimension of the texture map running along the U direction, and the Y (vertical) dimension of the texture map running along the V direction. This way, each pixel in the map appears exactly once, somewhere on the surface.



[Figure 10.19] The map radiates out from the mouth in mouth-centered geometry. Note that the squares on the texture are smallest near the mouth and stretched out farther away.

The way you construct a NURBS surface determines how your UV coordinates are distributed on your model. Figure 10.19 shows how a differently built NURBS model receives the map used in the last figure. This time, the model is built with the mouth as one pole of the geometry, so one edge of the map converges at the mouth.

Adopting Ptex

Ptex (per-face texturing) is a texture mapping system that provides a modern alternative to UVs through space. Ptex essentially creates UVs for each individual polygon in a polygon mesh or for each face in a subdivision surface. With Ptex, you don't need to choose a projection or spend any time setting up and editing UVs. Instead, your polygon mesh or subdivision surface is immediately ready to paint, so you can go straight to the texturing.

Since the Ptex format was introduced in 2010, it has been supported by a range of different programs. You can paint Ptex textures in 3D paint programs including Mudbox and Mari, and you can render them in programs such as Houdini, Mental Ray, VRay, and RenderMan.

Using Projections

A *projection* is a way to cast a 2D image into 3D space. The type of texture projection that is most similar to the way a movie projector projects film into a theater—which starts small at the lens and expands as it moves through space—is a *camera projection*. You can also choose from *planar, cylindrical, spherical,* and other types of projection.

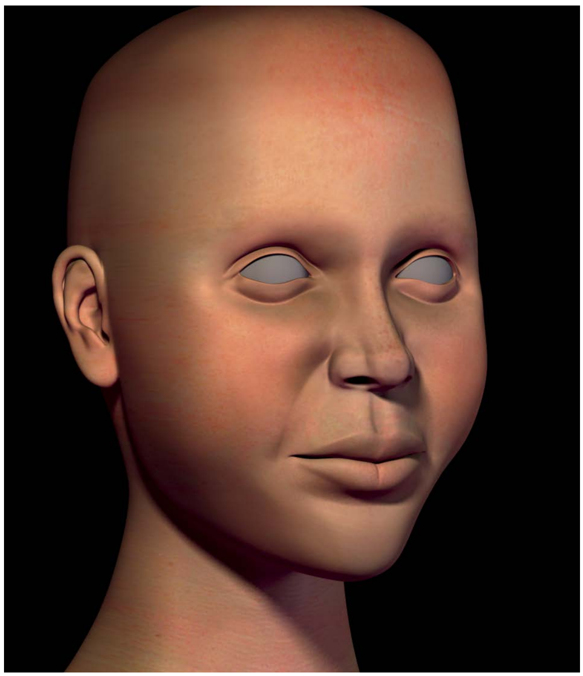
Assigning a texture as a projection is a more direct process than assigning UV coordinates via a projection. If you use a projection to assign UV coordinates, you are adding information to your model that you can then edit in a UV Texture Editor and bring with the model into a 3D paint program. When you assign a texture as a projection, you simply project the texture image itself through space, without the intermediate step of creating UVs that become a part of the model.

Compared to other ways of texturing a character, projections have an advantage: They are separate from the geometry, so even if you edit the model by adding or deleting some points, this will not cause the map to fall out of alignment.

Planar Projections

A *planar projection* casts your texture map onto a model so that the texture retains the same size and orientation when it is projected through 3D space. A planar projection projects a texture onto a model exactly the way you would see it if you painted it over a front, side, or other orthogonal view of your model.

Figure 10.20 shows a head textured via a planar projection from the front. Notice that the front of the head shows textural detail, such as freckles on the nose. One of the problems with planar projections is that a projection that perfectly covers the front of a model tends to stretch out or turn into streaks along the sides. You can see some streaking in front of the ear and along the left edge of the neck. The other problem with planar projections is that they go all the way through the model, so that a face projected onto the front of the head also appears on the back.

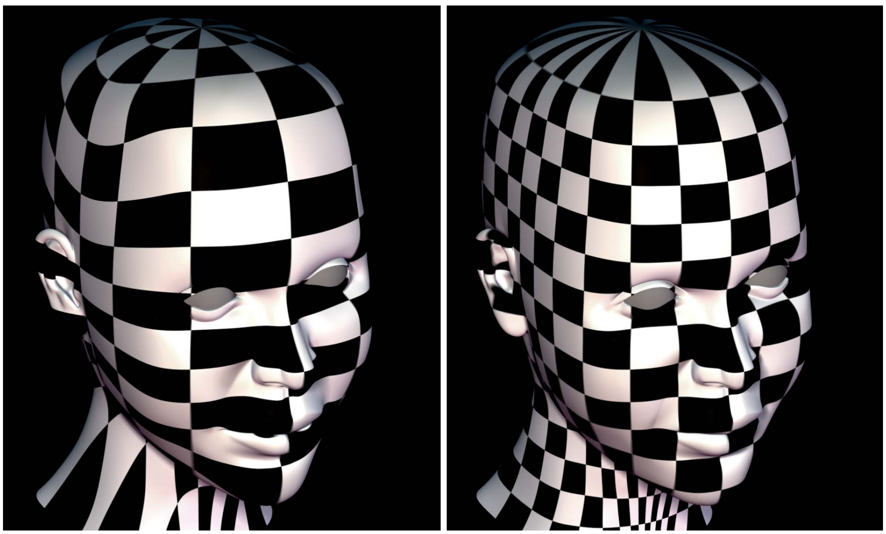


[Figure 10.20] A planar projection aligns with the front of the head but stretches out along the sides.

Because planar projections only do a good job of covering one side of a model, you often need more than one planar projection to texture a model all the way around. For example, you might paint one texture for the front of the head and another for the side of the head.

Spherical and Cylindrical Projections

Spherical and cylindrical projections cast an image inward from a sphere or cylinder positioned around the model. Figure 10.21 shows a head model with a grid applied in spherical projection (left) and cylindrical projection (right). You can see how the poles of the sphere come together to a point, projecting more of the map onto the top of the head (I'll talk more about texturing poles like this later in this chapter). This can be an advantage for cases where the top or bottom of the model is visible, but it may not matter if hair is going to cover the top of the head. The cylindrical map stretches out on the more vertical top surface of the head, but otherwise it covers the head more uniformly and consistently.



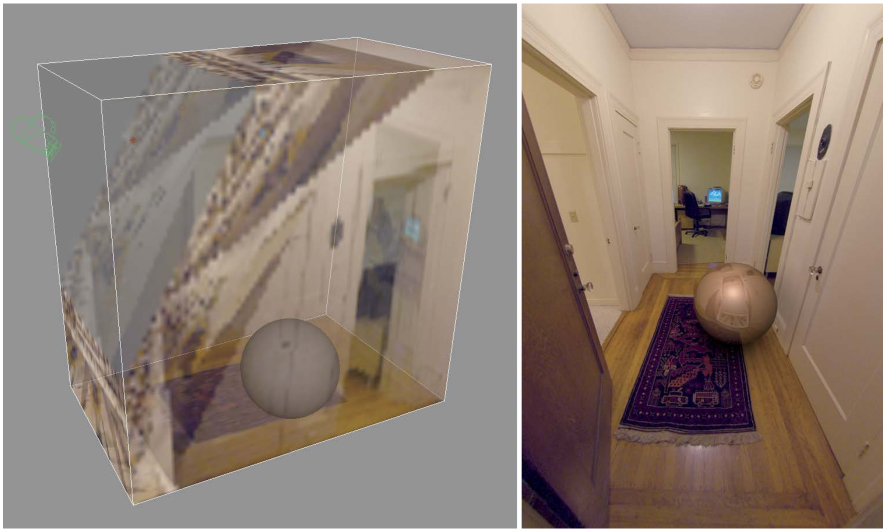
[Figure 10.21] A head mapped in spherical projection (left) and cylindrical projection (right). The cylindrical projection is more stretched out as you approach the pole on the top of the head.

These kinds of projections are a quick, easy approach to wrapping a texture map around round objects.

Camera Projections

Camera projection (also called *perspective* or *front projection*) is most similar to a movie projector or slide projector. It can exactly match the view of a specific camera in your scene. If you are painting or compositing together a digital matte painting, this kind of projection lets you paint over the actual geometry, as rendered, and then project new images into the scene in a way that exactly aligns with the geometry seen by the camera.

You can also use a camera projection to project your background plate into a 3D scene so that surfaces are textured with a view that exactly matches what the camera sees. This can help you make reflections match real-world environments. Figure 10.22 shows a camera projection of a real scene onto simple matched geometry. When it appears in raytraced reflections, it creates reflections on 3D objects that appear to reflect the environment into which they are composited.



[Figure 10.22] A cube is textured with the background plate using camera projection (left) so that computer-generated objects can realistically reflect the environment (right).

Other Projections

Different software supports different projection types. Many programs offer *cubic projection*, which combines several planar projections from different angles. Also useful are *custom* or *deformable projections*, which you can bend or edit to project the texture from just about any shape.

Creating Texture Maps

After you've chosen a basic strategy for aligning your texture with a model, you need to create the actual texture map. You can take a lot of approaches in creating maps. You can paint maps from scratch in 2D or 3D paint programs or scan or photograph real surfaces as a starting point for making maps. You can process your maps into tiling patterns that repeat or mask them out to layer onto surfaces as decals. Although painting texture maps is an art that can take years to master, a basic set of skills that add convincing details and dirt to your models will add enormously to your 3D work.

Using a 3D Paint Program

Painting texture maps in a 3D paint program is designed to be a simple, direct process. You generally start by importing a 3D model that has UV coordinates already assigned, or else you use Ptex. Once your model is in the 3D paint program, you can either paint directly onto the surface or paint onto a layer that gets projected onto the surface each time you rotate the model to a new angle.

You can use 3D paint programs to create complete, final texture maps. Some let you paint many kinds of maps, interactively previewing displacement, specular, and other attributes as well as color.

However, sometimes some 3D artists prefer having the control of seeing and working with textures in a regular 2D paint program, and they want to work with texture maps in a full-featured program like Adobe Photoshop. As a compromise approach, you can paint a very basic color map in a 3D paint program and then use that map as a background layer in a 2D paint program to guide the alignment of your final texture maps. For this purpose, any color map that shows the location of the main features and sides of the model is adequate if you are going to paint over it to create the final textures.

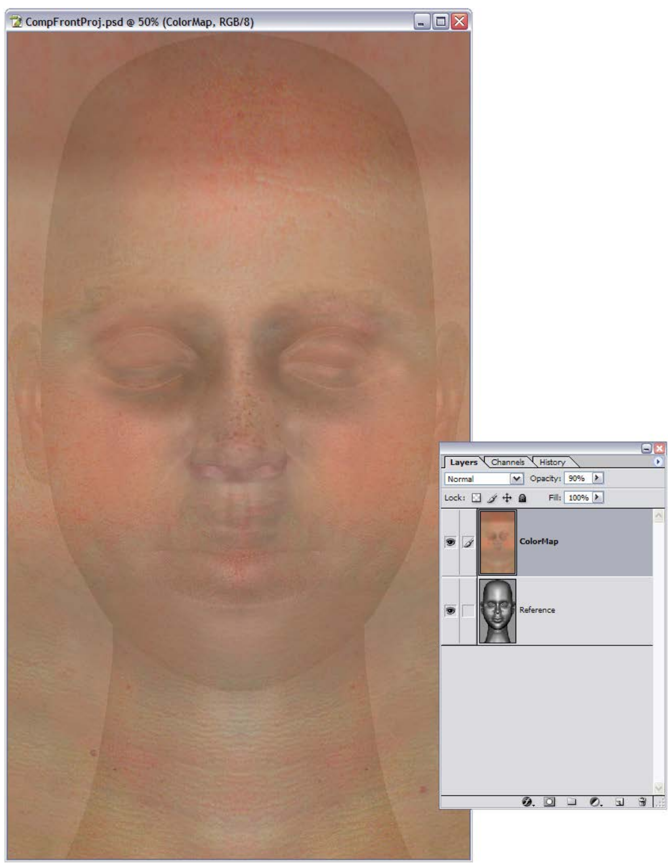
Using a 2D Paint Program

You can use just about any 2D paint program to create and edit texture maps. If you want to make texture maps that line up well with 3D models, then you need to bring a reference image of some kind into the paint program to use as a background layer to paint over.

If you are painting a texture map to follow the UV coordinates you assigned to a model, then create a screenshot of the unfolded model from the UV Texture Editor and paint or composite together your final texture on top of that. Using the screenshot image as a background layer for reference will show you where each main feature should appear within your texture map.

If you are painting a planar projection map, bring an orthogonal view of your model into your paint program for reference. For example, to paint a map that will be projected onto the front of a character's head, you need to bring a front view of the character into your paint program. The view you bring into the paint program doesn't have to be a screenshot; it can also be a high-resolution rendering that uses an orthogonal front view as the camera. Crop your reference image precisely so that the model appears with no extra space around it.

Once you have a reference image in your paint program, create a new layer, leaving the reference image in the background, as shown in Figure 10.23. This enables you to work in your paint program, painting or compositing together the texture map, and when you need to precisely position a freckle on the character's nose, you can see where to put it relative to the character's nose in the reference image. You can flatten the image and save it without the background when you bring your texture map into your 3D application.



[Figure 10.23] A front view of the model is used for reference in the background layer, while the color map (top layer, partially transparent) is composited on top.

Capturing Photographic Textures

You can boost to the level of realism in the texture maps you create by including portions of images photographed or scanned from real surfaces. Many professional texture painters use portions of digitized images within the maps they create. Even if you paint textures from scratch instead of using parts of the photographs directly in the map, you should collect high-quality reference images to better inform what you paint. Photographing or scanning real materials is an essential part of beginning your texture-mapping project.

You can find pictures of almost anything on the Internet. It's not illegal to collect pictures off the Web to study for your own visual reference or to view during discussions with the director and art director about what kinds of materials they like. However, it is best if you just look at and study images from the Internet; do not directly integrate them into your texture maps. Most images on the Internet are copyrighted. Even textures distributed on websites labeled as free textures might not really be rights-cleared from their original creators. And the images that are legally cleared for professional use are often not of the highest resolution, or they are stored as compressed JPEG files. To get the best-quality digital images that you can blend into your own texture maps, head out with your own camera and shoot your own pictures, or collect real material samples to scan in your own scanner.

There is a richness to real life that is worth going out to capture for each production, even when you are rendering a fictional subject. The actual subject you shoot may just be an analogy for what you are making. For example, you may texture the hull of a UFO with maps you derive from the panels of a city bus, or you may take close-ups of leftovers from a seafood restaurant to texture an alien creature.

Besides improving the quality and believability of your maps, photographic textures can also save you time when you're creating all of the maps you need for a particular production. No amount of digitizing eliminates the need for you to work at putting together textures in a paint program, but if you have great shots of real surfaces that look like what you are trying to create, you have a terrific starting point that can get you better maps in less time.

Shooting Tips

Not every photograph is equally useful when you're making texture maps. Try to follow these six tips when you go out with your camera to collect textures:

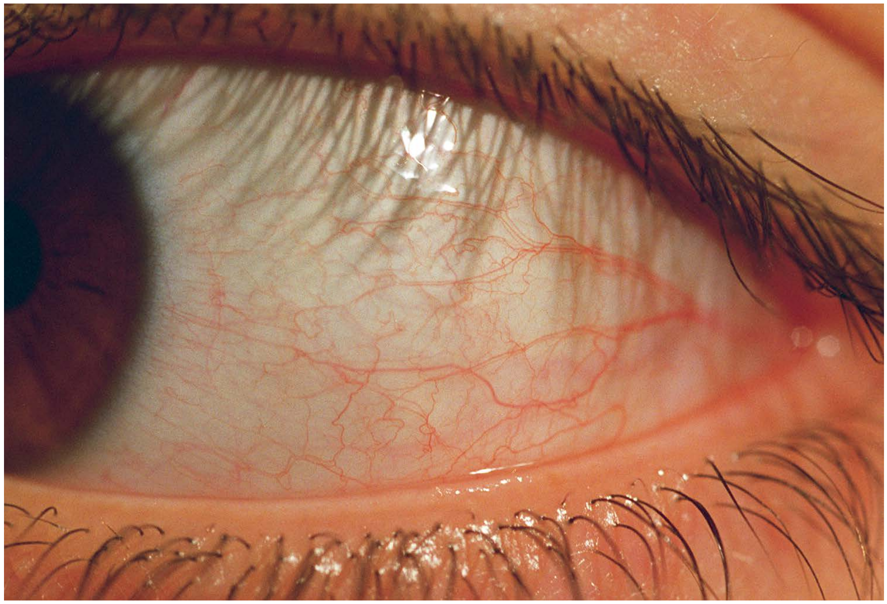
- To capture surface color textures, try to avoid highlights, shadows, or lighting variation in your photographs. Sometimes you'll find uniform lighting in the shade, with the sun behind your subject or when the sun is behind a cloud. If you can't avoid having a highlight, such as on a close-up of a person's eye, try to make it a small, concentrated one that is easier to retouch out, not a broad highlight that covers a big area.
- Shoot surfaces straight on, not from an angle. Take a moment to level and straighten the camera, and make sure that the horizontal parts of your shot are really horizontal. Taking pictures of ground surfaces can be tricky. You don't just want to aim the camera down and take a picture of your own feet or the shadow of your tripod, and you often need texture maps that cover areas several meters across; this type of scene is best shot from a few stories up in the air. You may need to stand on a bridge or pier to get a straight-down view of sand, dirt, or street surfaces.
- Try to reduce lens distortion. Many zoom lenses create barrel distortion when they are zoomed out and provide less distortion at medium or telephoto settings. If you have barrel distortion or pincushion distortion in your shots, correct for it in a program such as Adobe Lightroom before using what you shot to create texture maps.
- Shoot edges, not just the center of surfaces. Textures often change when you get near the edge of a surface, such as where a wall meets the ground. Focus on any area where two materials meet or intersect and get extra texture shots there. You may find later that some of these areas need their own maps.
- Collect texture maps in whole families or groups of related textures taken from the same location. For example, if you need oak leaves, collect several different leaves from the same tree so that your oak tree can have a natural variety in its foliage.

- Take some shots that are both wider and closer than you think you need. Close-up or macro photography of a subject reveals a different world of textures and patterns. Wider shots that show textures in context are vital when you are checking the scale of your textures. For example, during texture mapping, you need to decide how many bricks tall a house's walls are; a wide shot can be an ideal reference to help you get the texture scale right.

Following these guidelines sometimes takes some extra work and set-up time. For example, to achieve uniform lighting and a straight-on camera angle when shooting a banana texture, I had to peel it and lay out the peel on a cutting board, as shown on the top of Figure 10.24. To make a complete texture map, I covered the gaps using the cloning tool in Photoshop and cropped the resulting image, as shown on the bottom of the figure.

[Figure 10.24]
A banana texture shot from a peel (left) is cloned into a complete texture (right).

Shoot liberally when you are out photographing images for texture maps. There's no reason not to shoot a few dozen extra shots of whatever you find interesting in an environment. As you archive your shots from many productions, you'll find older images that you can use again and again. Figure 10.25 is an extreme close-up of the white of an eye that I have used in textures for several creatures and characters since shooting it.



[Figure 10.25]
Vein textures from an extreme close-up of an eye have been used in multiple projects.

Flatbed Scans

A flatbed scanner is a useful peripheral that modern 3D artists often overlook. Although scanners may not be as quick or as much fun to use as a digital camera, they can produce images with completely uniform lighting, an absolutely flat perspective, and perfect focus from edge to edge, so they provide ideal source material for creating texture maps.

Flatbed scanners can capture sharp images with high resolutions. If you scan a 10-inch by 10-inch fabric sample at 600 ppi (pixels per inch), you get a 6,000×6,000 resolution scan, or 34 megapixels. Some scanners label their resolution in dpi (dots per inch) even though you are actually counting how many pixels it scans per horizontal or vertical inch. Even the compact flatbed scanners you find built in to multipurpose color printers can often produce accurate 600 ppi or higher scans that rival well-focused shots taken with a professional-grade digital camera.

Flatbed scanners are great for scanning clothing, flat artwork, photographs, fabric and wallpaper samples, or any other materials that are flat and fit on the document-sized glass plate of the scanner. You can even take odd subjects, such as raw meat, kimchi, or a toupee, and lay them flat on a scanner and scan them at high resolutions. Figure 10.26 shows a hair texture I created by laying a hairpiece directly on a flatbed scanner.

Creating and Using Tiling Maps

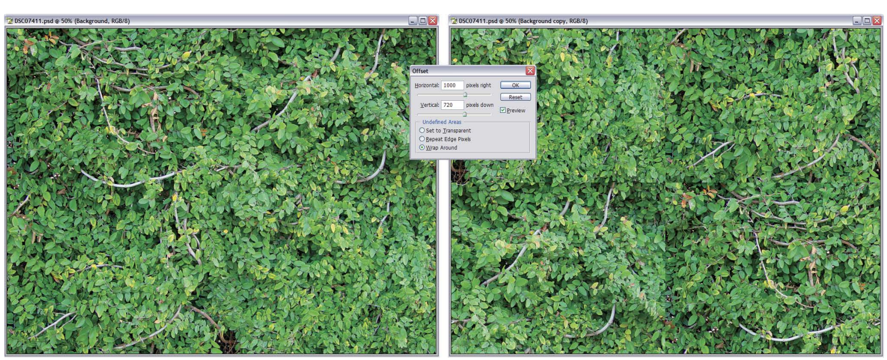
A *tiling map* is a texture map designed so that it repeats multiple times on a surface, and each repetition fits together seamlessly with the adjacent copies, like a pattern printed on wallpaper. The left edge of a tiling map lines up seamlessly with its right edge, and its top aligns with its bottom so that you can't tell where one repetition ends and another one starts, as shown in Figure 10.27. The white crosses indicate the four corners of the repeating shingle texture—if they weren't superimposed, you wouldn't be able to tell where the pattern repeats.

[Figure 10.26] This hair texture was scanned on a flatbed scanner.

[Figure 10.27] White crosses indicate corners of a seamless tiling map.

You can make a tiling map in almost any paint program. Begin by cropping a photograph or painted map so that the top and bottom, and the left and right, appear to be cropped at a similar point on the texture. After doing so, you should see nothing but a consistent texture within the frame.

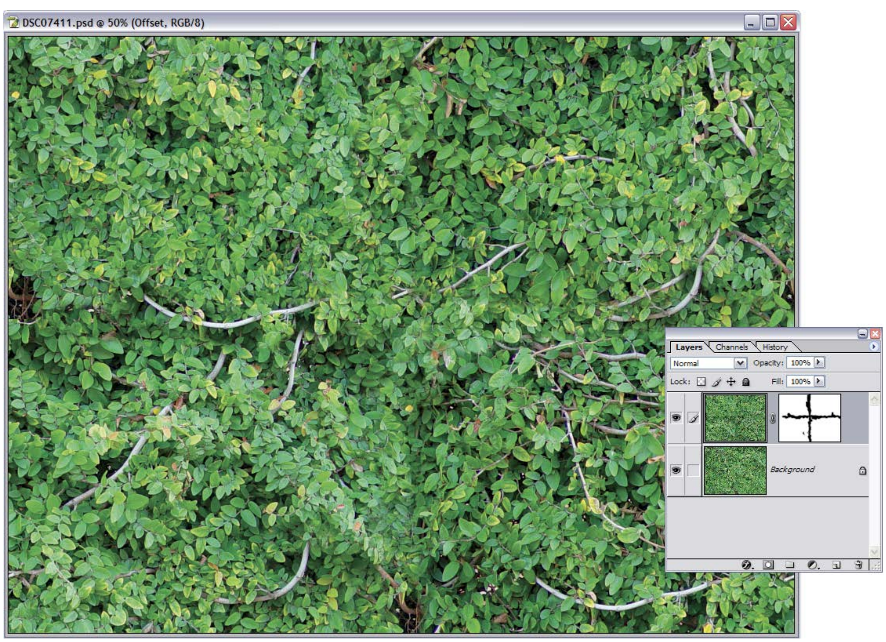
The next step is to offset the image so that you can see how the opposite edges fit together. Figure 10.28 shows how you can use Photoshop's Filter > Other > Offset command, starting with a cropped texture image (left). The image is slid by half its width to the right and down by half its height. Pixels that slide off one edge of the frame wrap around to appear on the opposite edge. The right side of the figure shows that, after the Offset operation, what used to be the four corners of the image are now all in the center of the frame. Notice that the reddish leaves that were in the upper-left corner are now just below and to the right of the center point. After the offset, opposite edges all fit together seamlessly in a tiling map; the only seams that need to be fixed are the ones now visible in the center of the image.



[Figure 10.28] The original image (left) is offset (right) to make it seamless at the edges and to draw the seams into the center.

There are several approaches to covering over the seams. You can use the cloning tool or healing patch, either of which copies other parts of the texture over the seams. The cloning brush enables you to paint the area being cloned with different levels of opacity, whereas the healing patch copies a whole region at a time but also color-corrects the region so that it blends seamlessly with its surroundings. You can also start with two copies of your layer, as shown in Figure 10.29, with the offset layer on top of the original, and then create a layer mask that hides the seams from the top layer. These techniques can be combined; start with the layered approach and then flatten the image and clone any remaining areas.

As a test to see if your map is truly seamless, roll the image around, using the Offset filter by different numbers of pixels horizontally and vertically to see where seams or discontinuities remain. For a test that actually shows you multiple repetitions, scale down the map to a smaller size, define it as a pattern, and use that pattern to fill the background of a larger image file.

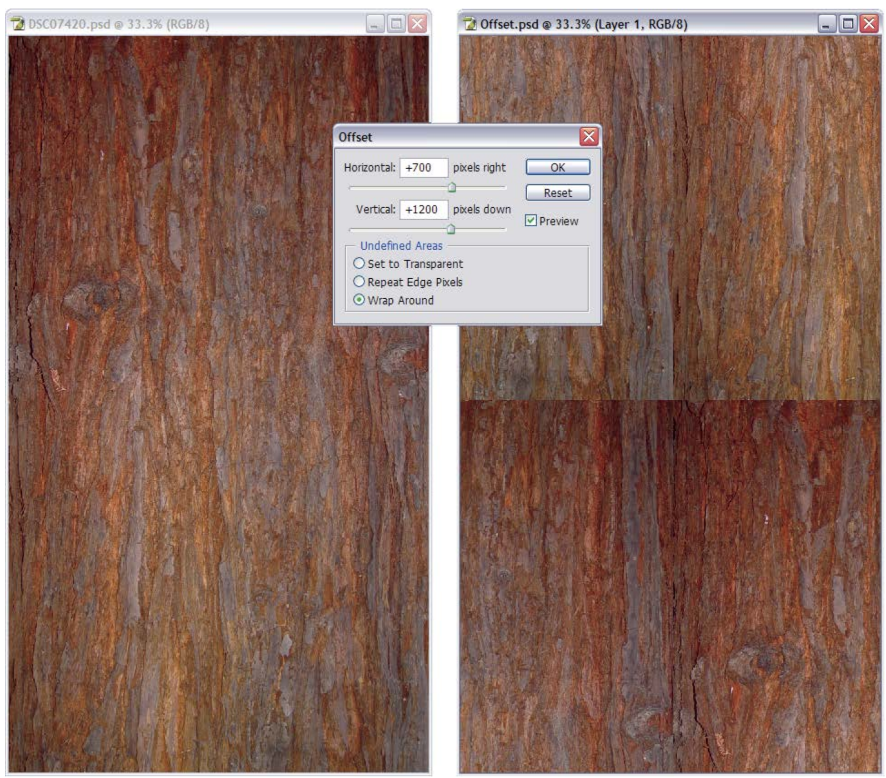


[Figure 10.29]
Seams in the top layer are hidden by painting a layer mask.

Correcting Luminance and Color Shifts

If you are starting with photographed elements, you need to correct any overall shifts in brightness or color when you are preparing a tiling map. Often a photograph is brighter in the center than on the edges, or an overall gradient shifts one side to a different color than the other. Even if a map like this is made seamless at the edges, when it is applied with many repetitions in 3D, you would see the brightness or color shifting back and forth with each repetition across a surface, creating a visible pattern from a distance.

Figure 10.30 shows a picture of a tree that has been cropped (left) and offset (right) to prepare it so it can become a tiling tree bark texture. An overall shift in brightness and color ran across the original image. The offset makes this especially visible along the center seam.

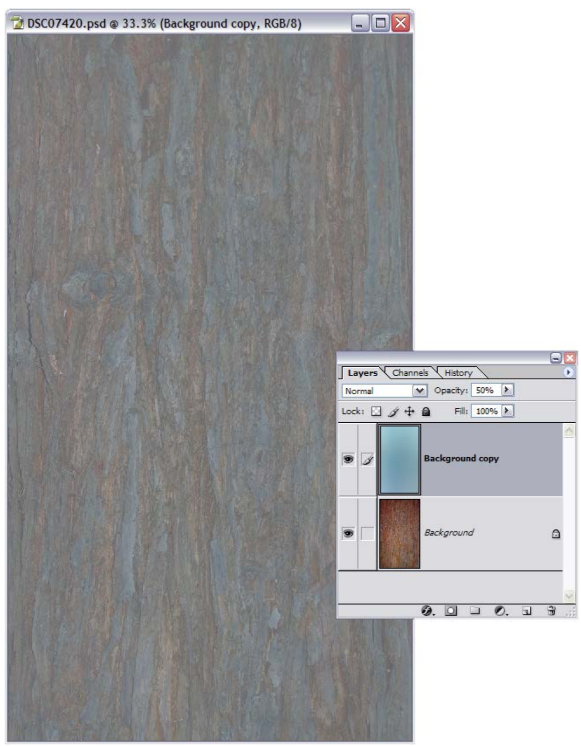


[Figure 10.30] Color and luminance shifts are revealed by the Offset filter.

To correct overall color shifts, one approach would be to use the gradient tool to paint gradient selection masks and then adjust the levels on different parts of the image.

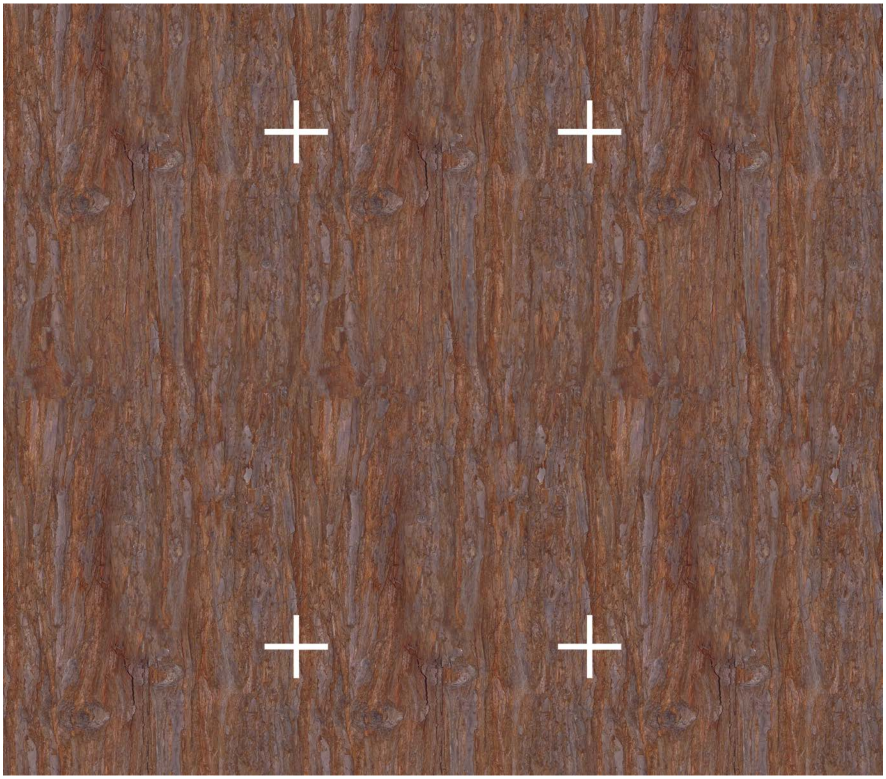
Another great tool for equalizing the brightness and colors in a map is the Filter > Other > Highpass filter. If you apply a Highpass filter before you do the offset, you can cancel out all of the overall variation in a layer while preserving the detail and texture smaller than a specified blur radius. Highpass is a very heavy-handed filter, which greatly reduces your image's overall contrast. After using Highpass, you'll need to adjust the levels of red, green, and blue to restore the color and contrast of your texture.

If you want more control than a Highpass filter offers, you can create a similar result for yourself. Make a copy of the layer that needs its colors and tones equalized. Blur the copy using a big enough radius to hide the texture and then invert the layer. If you mix the inverted layer at 50% opacity, then you have re-created the results of a Highpass filter, as shown in Figure 10.31.



[Figure 10.31] Simulating a Highpass filter with multiple layers gives you more control over the process.

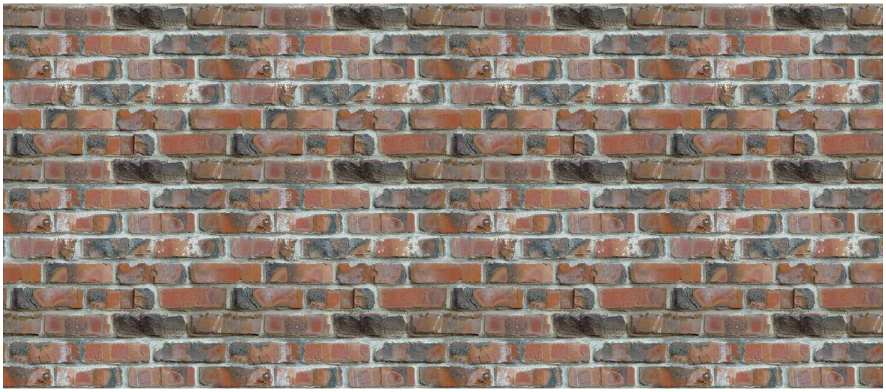
After building a Highpass-like result for yourself, you have the freedom to adjust the opacity of the top layer to less than 50% for a more subtle effect, or you can use a blending mode such as luminance if you don't want to cancel out the colors. After you have merged the layers, you will still probably be left with a low-contrast image, and you may need to increase the contrast of the image or use Auto Levels in order to bring out the contrast and variety to the texture map. The corrected map repeats three times across Figure 10.32 without shifting brighter or darker.



[Figure 10.32] After luminance equalization, the map repeats without shifts in tone. White crosses mark a single instance of the map.

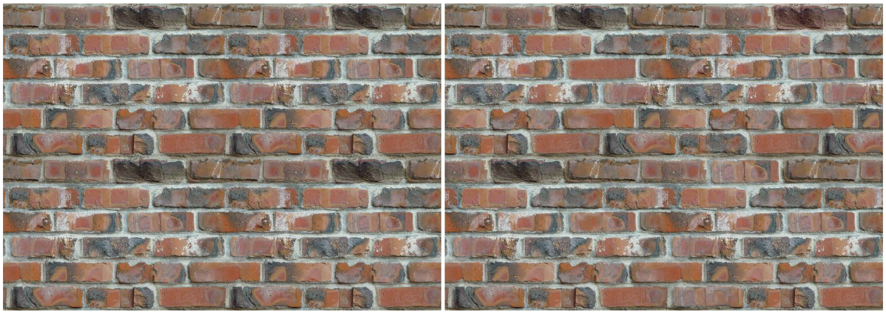
Expanding a Tiling Map

When you look at several repetitions of a tiling map, sometimes you see distinctive features that repeat too often, calling attention to the repetition of the map. For example, Figure 10.33 shows a tiling map of a brick wall with a variety of discolored bricks. It repeats four times across the figure, and the colored bricks form an obvious repeating pattern instead of looking random.



[Figure 10.33]
Repetition is too obvious in this tiling brick texture.

One way to reduce this problem is to expand the texture map. To do this, create a new image with twice the width and twice the height of the original. Paste four copies of the original into the new image so that they fit together to form four repetitions of the pattern, as shown on the left side of Figure 10.34. You can flatten all the copies onto the same layer. At this point, use the cloning tool (or lasso, cut, and paste) to move around some of the bricks. You may also want to change some of their colors, or anything else that breaks up the repetition of the pattern. I've shown the results on the right side of the figure. This new, expanded tiling map can cover larger areas without visible repetitions.



[Figure 10.34]
Four repetitions of a tiling texture are pasted together (left) and then edited to make a larger texture that repeats less often (right).

Before you start expanding a map, however, it's a good idea to test-render it in your 3D scene. Sometimes repetitions appear obvious to you when you preview the tiling in a paint program, but when a surface is broken up with different illumination and shadows, and parts of it are covered up by other texture maps or other objects, you can get away with a lot more repetitions of a map without a pattern becoming too obvious.

Horizontal and Vertical Tiling

One of the great advantages of using a tiling map is that you can adjust how many times it repeats horizontally and vertically across a surface. This makes it easy to get the scale of your textures just right. If someone expects bricks in a brick building to be a certain size, then your texturing only looks convincing if you get the scale of the bricks just right compared to the building. With tiling maps, you can fine-tune the scale after you create your maps, instead of needing to work out the texture scale while initially planning the maps.

Maps don't always need to repeat both horizontally and vertically. You can make some maps that are designed to tile only in a horizontal direction or just in a vertical direction. Figure 10.35 shows a map covering the lower edge of a cinderblock wall. It tiles only horizontally and is designed to run along the lower level of a surface.

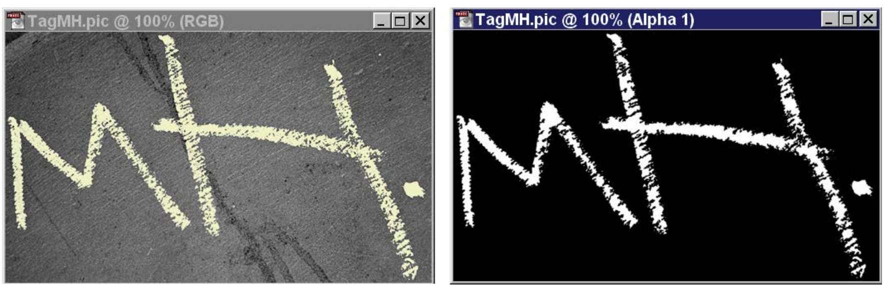
[Figure 10.35] Making a more realistic transition from the wall to the ground, this map tiles horizontally but not vertically.

Decals

In texturing a detailed or realistic object, sometimes you need to layer together more than one texture map. Sometimes you may assign some of the maps to different attributes, such as your color map, your bump map, and so on. You can also apply some of the maps to different regions of your model, for example, if you use separate maps near edges of a surface. If you want to add a unique detail to a specific position of an object, even in a region where you already have other textures applied to the same attribute, then you may want to create a decal.

A *decal* (sometimes called a *layered texture*, *stencil*, or *label map*) is a texture map that has been masked-out for layering on top of other textures, so it can add a unique detail to a specific position on an object. As shown in Figure 10.36, creating a decal usually requires both a color image (top) and a mask (below). The color image contains the texture of the decal area, and the mask contains white areas where the decal image should appear and black where the background material should be visible. The mask is usually stored in the alpha channel of the color map but may be a separate image file.

When using alpha channel layering, you can layer many decal maps on top of one another, building up more detailed scenes, as shown in Figure 10.37.

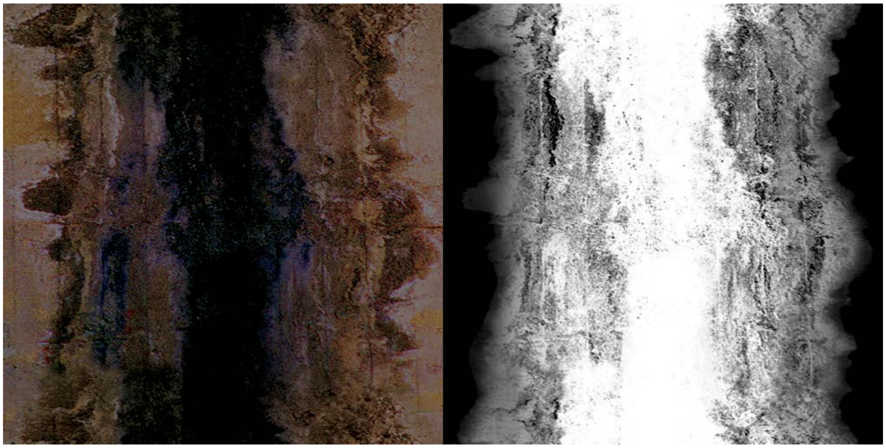


[Figure 10.36] A decal map is created along with an alpha channel to control how it is layered over other maps.

[Figure 10.37] Arrows indicate some of the locations where the map from Figure 10.36 is used throughout the scene.

Making Realistic Dirt

One of the most useful purposes for decals is to help you add dirt to your models. You can make maps that represent dirt, smudges, and other surface imperfections and layer them on top of other textures. Figure 10.38 shows a map of a water stain (left), designed to repeat vertically, masked with an alpha channel mask (right). A vertical map like this helps break up and add variety to a horizontal map like the one in Figure 10.35.



[Figure 10.38] This vertically tiling water stain texture (left) is layered via an alpha channel (right) over other textures.

When you need to make something dirty, don't just superimpose random noise on it. Instead, choose dirt maps that add specific, motivated detail to your objects. Think through the story behind all the stains and imperfections on a surface—something has to cause any dirt, scratches, or stains that you see. Here are a few examples of motivations that can influence your design and how you position dirt maps:

- Scratches don't appear randomly on a floor; they are more likely to appear in an area that gets scraped when a door opens.
- Carpets are most worn in the path where people walk.
- Mildew grows darkest near cracks and corners.
- Water leaves stains in the path where it drips down a surface or where its surface reaches a high-water mark.

Figure 10.39 shows a photograph of rust stains on a cement wall. Notice that they are not random; the streaks are concentrated below metal posts, the source of the rust. This kind of thinking is especially important when you are trying to make a fantasy world or imaginary environment believable. You need to imagine the whole history of what has happened in an environment in order to invent convincing dirt for it.

[Figure 10.39]
Rust stains appear where water drips from metal fixtures.

Also, dirt isn't all dark. Dirtying up your surface doesn't just mean darkening it or making it less saturated. Some kinds of dirt—such as dust, scratches, water stains, and bird droppings—make your surface lighter in color. Rust, rot, and fungus can even add richness and color to an aging surface.

Making Maps for Multiple Attributes

You often need to create texture maps for multiple attributes of a surface. For example, you may need maps for the color, specularity, bump, and displacement of a surface. There's more than one strategy for developing a complete set of texture maps that will work together when you are mapping multiple attributes.

Color First

The color first strategy just means you start with a color map before you make any of the other kinds of map that complement it. Textures derived from photographs or scans tend to fall into this category.

Sometimes when looking at a color map, you may decide that lighter tones in the map tend to appear where a surface is higher, and darker areas appear where the surface is lower, so you can let your color map double as a bump map. If dark features on your color map represent things that actually bump higher, then you can invert your color map to make a bump map. For example, if you have a brick wall with white mortar between dark red bricks, you need to invert it so the mortar appears darker before it can make a good bump map for the wall. Often, though, you will see some details in your color map that shouldn't cause any change of height at all; retouch these out of the map before you use it as a bump. Your bump map should only emphasize the features that would bump visibly in or out from the surface.

Sometimes a specular map can start as a black-and-white version of your color map. But you need to think about whether each feature in your color map really affects your specularity. Many features do not, if they are simply color changes without any change to the shininess of the surface. If a feature shouldn't affect the specularity of the surface, then you should remove it from the image before using the image as a specular map.

Displacement First

Painting *displacement first* (also called *bump first*) is a strategy where you paint the basic structure of a map before you assign the color. For example, if you are creating the look of a stained glass window, you can start by painting the structure of the metal frames that go around the individual pieces of glass. You can then use this map as a bump map. You can also use it as the starting point for painting the color map, which you can make by filling in the areas between the metal frames with different colors. You can create a transparency map in the same way you created the color map, but with solid black for the metal frames between the colored glass. As a result, a complete look that varies several attributes is painted starting with the basic structure contained in the bump or displacement map.

Painting in Layers

One of the great advantages of painting your textures from scratch, instead of processing photographs or scans into texture maps, is that you can paint different elements of your textures onto different layers in your paint program.

As an example, when you are painting a skin texture, you can paint features such as freckles, pores, hairs, dirt, and tan lines onto different layers. Because you paint each of these features on separate layers, this makes it vastly easier to create other types of maps based on your color map. When you create a bump map, you can decide which features will be a part of a bump map and which will not. For example, freckles only represent a color change, so you can omit a freckles layer from the bump map. Pores, however, are depressions into the skin, so you turn them into dark spots on the bump map. The razor stubble on a man's chin bumps outward, so you can turn that layer white and add the white dots to the bump map.

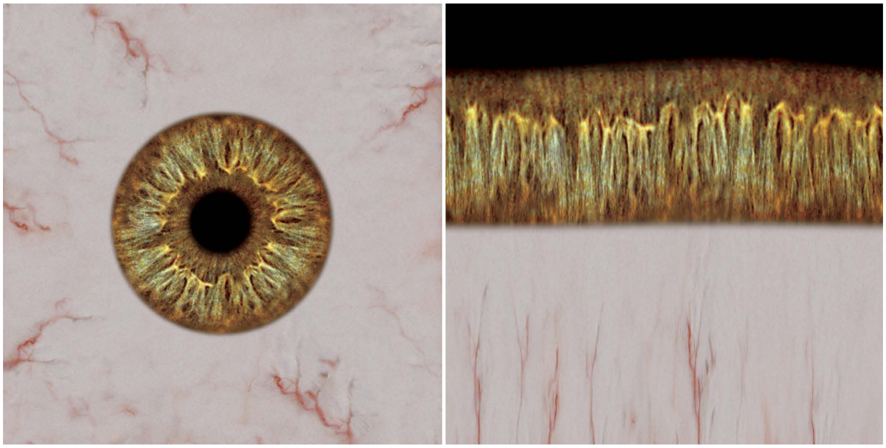
When painting dirt into textures, sometimes you can make custom brushes by starting with photographic images; these custom brushes can add realistic detail to the layers you are painting. Dirt tends to influence several of the texture maps on a surface. Of course dirt can change the color of a surface, but it also can make the surface less shiny, so it can be represented as a darkened area in a specular or reflectivity map. Sometimes artists add the texture of a dirt pattern or stain to a bump map as well, and if a surface is transparent, then dirt also tends to create a darker area (less transparent) in the transparency map.

You need to decide in which texture maps every feature, every piece of dirt or scratch on a surface, should appear. Usually studying a real example of the material you are trying to re-create is the best guide to this. For each surface feature, ask yourself how it influences the shininess, the bump, the transparency, or other attributes of a surface before you put that feature into additional texture maps. When you have features appear in several types of map, the different maps you apply to the surface can reinforce each other to create a more believable look for your surface.

Texturing Poles

Poles are places where an entire side of a texture map gets pinched into a single point. Poles occur at the top and bottom of a spherical projection and also happen in UV coordinates at the geometric poles of a NURBS surface.

You can use the Photoshop function Filter > Distort > Polar Coordinates to switch a map from having a pole in the center of the image to having the pole represented as one edge of the image. Figure 10.40 shows a texture map of an eye before the filter is applied (left) and after (right). If the eyeball had been off center, then instead of horizontal lines, the edge of the iris would appear wavy.

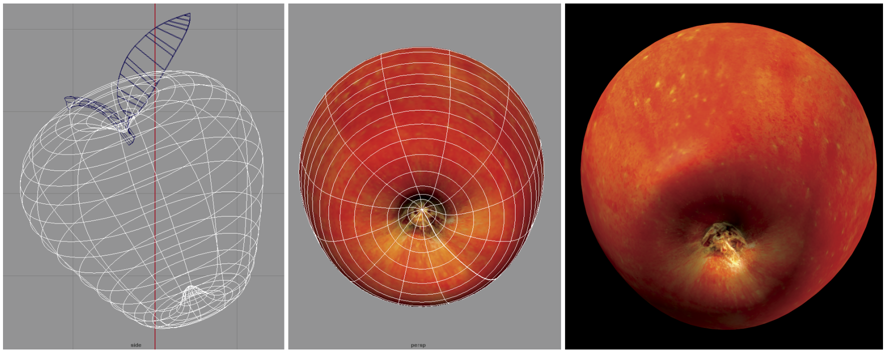


[Figure 10.40] An eyeball texture map made for planar projections (left) and designed to radiate out from a pole (right).

This conversion optimizes an eyeball texture map so that it can be applied according to the UV coordinates of a spherical eyeball with the pole facing outward. Another benefit to the conversion is that it makes it easier to retouch a highlight or eyelash out of a picture, because you can just clone or copy other parts of the texture from the left or right of the area. If you want to work this way only temporarily, the filter can also convert in the opposite direction to reverse this effect.

Texturing an apple is another situation in which parts of your map cover poles. A NURBS apple, shown on the left side of Figure 10.41, has poles at the top and bottom. The geometry converges at the bottom pole to a single point (center) but needs to be textured without the texture appearing to pinch together, as rendered on the right of the figure.

To cover the apple, I used a map that stretches out at the top and bottom poles. Figure 10.42 shows a photograph of the bottom of an apple (left) and the bottom edge of the texture map I created with the Polar Coordinates filter (right).



[Figure 10.41]
A NURBS apple (left) features a pole at the bottom (center) that can be seamlessly textured (right) with an appropriate map.

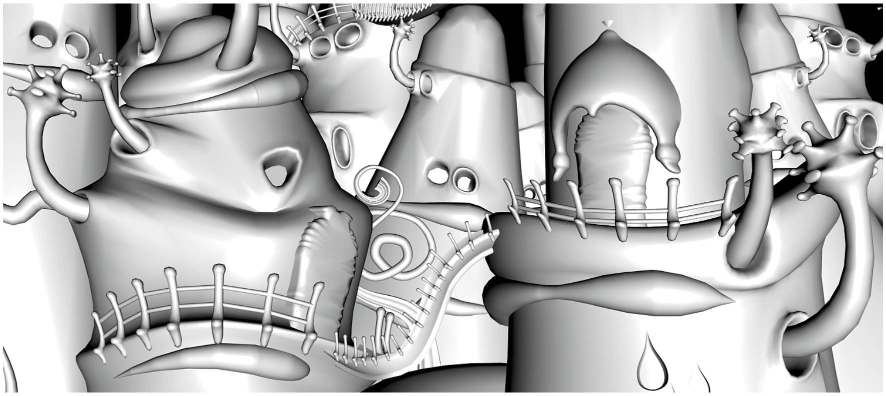
[Figure 10.42]
The source picture of the lower apple (left) is converted to polar coordinates (right) and integrated into the bottom edge of the texture map.

Painting Stylized Textures

For some projects, a non-photoreal visual style dictates that you should paint maps from scratch, without letting real-world imagery intrude on your scene. To take this approach, it is especially useful if you have experience drawing and painting in traditional media. While you may create hand-painted maps in almost any visual style, they are especially important for fanciful and illustrative renderings. Figure 10.43 was textured with hand-painted maps that Eni Oken created in a 2D paint program.

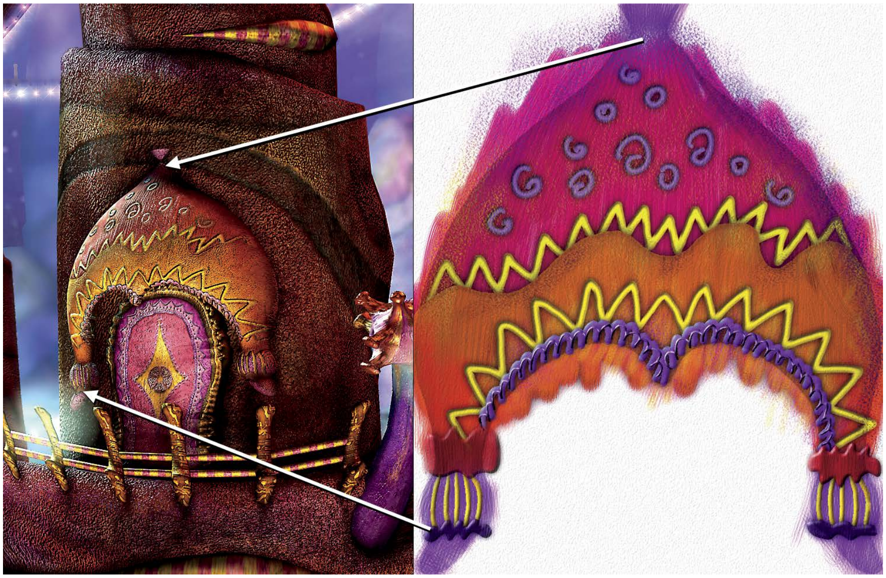
[Figure 10.43]
Hand-painted texture maps maintain the illustrative style of a fanciful scene Eni Oken (www.enioken.com) created using 3ds Max.

The models underneath your texture maps may be very simple, like the buildings in Figure 10.44. The texture maps add richness to the scene while preserving the fanciful, ornamental style of the modeling.



[Figure 10.44]
The untextured buildings are simple forms.

Figure 10.45 shows one of the texture maps being applied to the awning of a building. The map matches the shape of the geometry, and adds appropriate colors and ornamental details to the awning. The artist carefully chose colors in the map to integrate with the scene's color scheme, using the same purple, yellow, and orange tones that are picked up in many other surfaces.



[Figure 10.45]
Ornamental texture maps are added to the color and bump of the buildings.

Light and shadows are painted into these maps to reinforce the lighting in the scene or to simulate details and dimensionality that never existed in the underlying geometry. Note how the door texture map in Figure 10.46 already contains shading and shadows. In the final rendering, these are added together with the shading and shadows of the geometry. Although this is a cheat that can detract from the realism of animated productions, some artists do cheat this way, especially when they are creating non-photoreal style renderings.

[Figure 10.46]
Color maps for the stylized rendering already contain shading and shadows.

Texture Map Resolution

How high does the resolution of your texture maps need to be?

Start by thinking about the resolution of your final output. For example, many productions are rendered for motion picture projection or HDTV delivery at a resolution of 1,920 pixels across by 1,080 high. There is also growing interest in 4K resolution television distribution and movie projection, which uses a resolution of 3,840×2,160.

Some texture maps can be lower resolution than your rendered frames. For example, if your texture map covers a small object that fills only half the frame, then it needs to be only half the output resolution. If a map repeats three times across a surface, then the map needs to be only a third of the resolution. This is why you can often use texture maps that are only 1,024 pixels across, and they can still look sharp and detailed when seen in a feature film.

However, there are times when you need higher-resolution maps. If a map wraps all the way around a character's head, then only a portion of the map covers the character's face. In a full-screen close-up, the face itself needs to have enough detail to fill the screen. If a texture map covers a mountain or a whole planet, then close-ups of a small part of the object require maps many times the resolution of the screen—although in extreme cases you probably need to use a separate map for close-up areas.

Most graphics cards and many renderers work most efficiently with texture maps whose dimensions are a power of two, such as 512, 1,024, 2,048, or 4,096. In fact, some software and hardware internally scales up maps of other dimensions to the next power of two. Table 10.1 shows some of the most common texture map resolutions and their RAM use for a four-channel file.

[Table 10.1] Texture Map Memory Use

MAP RESOLUTION	MEMORY USED
512 × 512	1 MB
1,024 × 1,024	4 MB
2,048 × 2,048	16 MB
4,096 × 4,096	64 MB
8,192 × 8,192	256 MB

When you digitize a map or work with it in a paint program, try to design your textures at a higher resolution than the final maps you'll need in 3D. For example, if you are going to use a 1,024×1,024 map, start with an original file at least 2,048 pixels across. Many steps in a paint program, such as rotating the image, can slightly lower the image quality. If you start with a higher resolution than you really need, then all of your image editing can appear seamless after the map is scaled down to its final size.

If you start with a high-resolution map, you can always make a scaled-down copy if you want a smaller file that uses less memory. However, your texture maps have a limited amount of detail. If a texture map is too low a resolution for your needs, you can't gain detail by scaling up the map—scaling up a low-resolution map only produces a softer image.

The only kind of resolution that matters in your completed texture maps is the size in pixels. The dpi (dots per inch) or ppi (pixels per inch) setting on a completed texture map won't matter in 3D, so you can leave it at whatever value happens to appear there; 72 and 96 are common defaults. If you are using a flatbed scanner with an adjustable dpi or ppi for scanning, then use whatever setting provides you with the number of total pixels that you need to create your map.

Using Procedural Textures

A *procedural texture* is an algorithm in the computer that can generate a colored pattern based on input parameters without needing to load an image file.

Procedural textures are an alternative to creating texture maps. They add textural detail to your models that are calculated while you render, without you needing to apply any texture maps.

Compared to texture maps, procedural textures have some advantages and some disadvantages.

Resolution Independence

Procedural textures are *resolution independent*. When you make a high-resolution rendering or zoom in to an extreme close-up, a procedural texture can reveal more and more detail, and never becomes blurry like an overstretched texture map. Most procedural patterns can be mathematically derived at any scale, although some have a setting for the number of iterations or levels of detail to be computed.

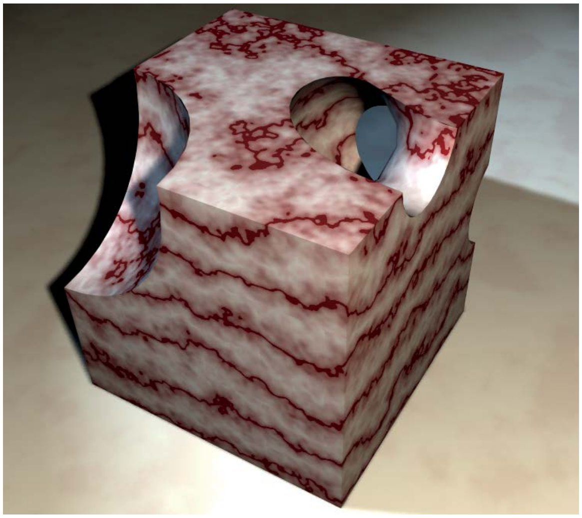
Due to their resolution independence, procedural textures are well suited for covering broad areas of a scene, such as for a landscape or terrain. A procedural texture can cover an infinitely large ground surface without ever repeating.

However, be warned that the resolution independence of procedural textures is not a guarantee that the close-up appearance of a texture will meet your needs or expectations. No matter which kind of texturing you use, when you are designing a 3D scene to appear realistic and detailed in extreme close-ups, you must study how your subject appears at very close range, and follow this up by test-rendering and developing appropriately detailed textures.

3D Textures

Some procedural textures are *3D textures*, also called *solid textures*. Instead of being two-dimensional like a texture map, a 3D texture is a procedure that produces a pattern based on a three-dimensional position in space. You can model any shape object, and a 3D texture can be applied evenly to every part of it. For example, the marble texture in Figure 10.47 wraps seamlessly from one side of the cube to another, and even into holes that cut through the object. This would require effort to set up if each surface were textured with a texture map, but it happens automatically with 3D textures.

Many programs also support 2D procedural textures, which you can project or apply to UV coordinates, just like a texture map.



[Figure 10.47] A 3D procedural texture uniformly textures all surfaces within its 3D space.

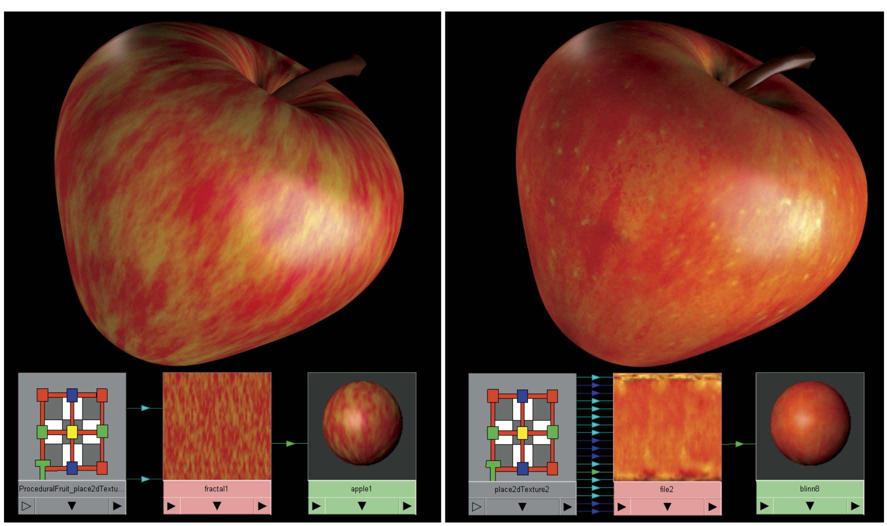
Animation

Procedural textures are controlled by adjusting numeric parameters. You can animate parameters so that when they change over time, your texture changes in all sorts of ways. Animated procedural textures can produce unique appearances that you can use in visual effects, transitions, or even as abstract animated backgrounds to appear behind titles and logos.

Appearance

Although features such as resolution independence, seamless 3D texturing of complex models, and animatable parameters are great technical advantages, make sure you consider the final appearance you can create when you choose an approach to your texturing. Most 3D programs come with a set of general-purpose procedural textures that add different types of noise and patterns to a surface. If you are working on putting together scenes very quickly and want to create textures without needing to leave your 3D program, then assigning a general-purpose procedural texture to a surface can be a quick way of putting colored patterns onto different objects.

General-purpose 3D textures seldom look exactly like what you are approximating with them. For example, a procedural wood texture can look somewhat like wood, or perhaps more like a simulated wood pattern, but it isn't likely to look like antique oak furniture, or softwood cedar, or maple, or anything more specific. If you care about getting the look exactly right, you should probably create texture maps instead of relying of procedural textures. Figure 10.48 shows a comparison between a general-purpose procedural texture (left) and a custom bitmap texture (right); although the procedural texture might not look as realistic, it was created in a tiny fraction of the time.



[Figure 10.48] A simple procedural approach to texturing an apple (left) is faster to create than a bitmap texture (right), but it may not look as much like an apple.

Artists working on quality-oriented productions sometimes use procedural textures in combination with texture maps. For example, you create a texture map for an apple based on photographs but map the specular color of the apple with a procedural noise to simulate variable amounts of shininess on the surface. You may find procedural textures most useful when you use multiple procedural textures together and combine them with texture maps.

Another way that procedural textures are used in high-end productions is through custom shader development. All large studios and many smaller companies employ programmers or shading technical directors who write custom shaders. *Custom shaders* are shaders written for the needs of a specific production, and they may include procedural textures developed to achieve specific looks for a particular type of surface. Although writing custom shaders can take longer than painting a texture map, these shaders are useful because they can cover an entire planet or other large or varied objects with nonrepeating patterns.

Baking Procedural Textures into Maps

Some programs contain functions that can bake procedural textures into texture maps for you, which means converting them into regular texture map images. In Maya, for example, the Convert to File Texture function in Hypershade converts any procedural texture into a bitmap texture.

If your software doesn't have any baking or conversion functions, a universal way to turn procedural textures into texture maps is to apply the procedural texture to a uniformly bright flat surface and make a rendering of the surface. You can then apply the rendered texture as a map on different objects.

The main reason to convert procedural textures into bitmaps is to gain additional creative control. The procedural pattern might be a main part of your texture, but you may want to paint in different details around particular areas, such as stretching out part of the texture around a joint, or adding unique wounds or imperfections to a texture.

Although the general-purpose procedural textures that ship with most 3D software may not meet all of your texturing needs, using them can sometimes be a great way to start when you are painting a more detailed or interesting texture. Instead of using the random noise function in your paint program, bake textures in your 3D program to give yourself access to a much bigger library of noise and procedural patterns that you can use as starting points in painting maps. Even when a procedural texture is too simple or regular to use as the final texture, you may still find it a useful starting point to bake into a bitmap and bring into your paint program.

Another reason to convert procedural textures into bitmaps is to move textured models between different programs. Different software will have different procedural textures available, but every program should be able to render a texture map.

In some cases, baking procedural textures into maps may make some textures render more quickly, especially when you are rendering with motion blur. If you're converting entirely as a performance optimization, be sure to test-render a frame of the moving object with the procedural texture first, and then render the same frame again after baking the texture into a map. In some cases you experience a speed-up, but check the actual rendering times to see if it's significant enough to make this worth doing.

Both procedural and bitmap textures have their place in production. Maps may be more popular because they offer additional control and random access to coloring specific points on an object, but never overlook the possibility that a procedural texture may be the best tool for a particular job.

Great texture maps can come from almost anywhere, whether they are baked from a procedure, photographed, scanned, or painted. The keys to successful texture mapping are learning to visualize what you want to create and being flexible about how you accomplish your goals.

Exercises

1. A terrific texture mapping project for your portfolio is to study and reproduce a small outdoor area in 3D—it can be the area around a tree stump, a fire hydrant, or a potted plant, or any space about a meter or two across. Photograph it with a digital camera. Shoot some wider shots from multiple angles for modeling reference and then an extensive set of close-up pictures for texture map development. Model and texture map that area to reproduce it as completely as possible.

2. You may have downloaded one of the Lighting Challenge scenes from www.3dRender.com/challenges already. Download one if you haven't yet and try texturing it. Note that challenges that are close-ups on smaller areas, such as the Halloween challenge, tend to be easier to texture because a smaller number of surfaces are visible that need texture maps.